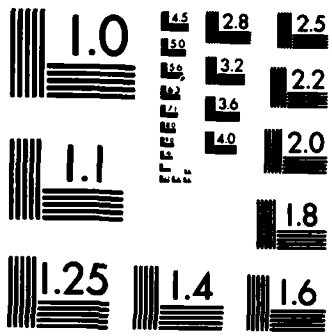
AD-A157 392 AISIM (AUTOMATED INTERACTIVE SIMULATION MODELING SYSTEM) OVERVIEW REVISION 1(U) MITRE CORP BEDFORD MA H P SCHULTZ MAR 85 MTR-8707-REV-1 ESD-TR-85-149-REV-1 1/1

UNCLASSIFIED F19628-84-C-0001 F/G 9/2 NL

END



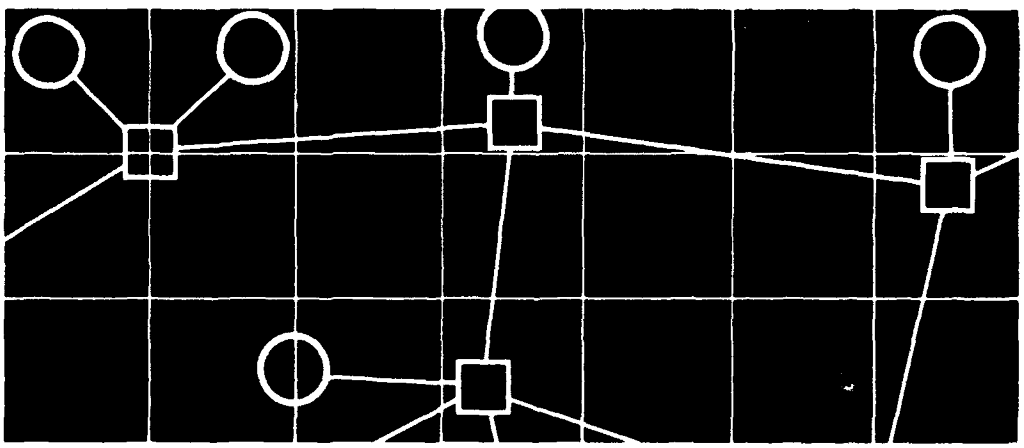
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AISIM Overview

April 1985

ASIM

Automated
Interactive
Simulation
Modeling
System



Prepared for Deputy for Acquisition Logistics and Technical Operations, Electronic Systems Division, AFSC, United States Air Force, Hanscom Air Force Base, Massachusetts.

MITRE
Bedford, Massachusetts

MTR 8707 REV 1
ESD-TR-85-149

Approved for public release; distribution unlimited

When U.S. Government drawings, specifications or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Do not return this copy. Retain or destroy.

REVIEW AND APPROVAL

This technical report has been reviewed and is approved for publication.

Accession For	
NTIS GRA&I	X
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A/1	

N. Ann Kuo

N. ANN KUO, 2Lt USAF
Project Manager,
Requirements Analysis

William J. Letendre

WILLIAM J. LETENDRE
Program Manager, Computer Resource Management Technology

FOR THE COMMANDER

Robert J. Kent

ROBERT J. KENT
Director, Computer Systems Engineering
Deputy for Acquisition Logistics and Technical Operations

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS
None

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release; distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
MTR-8707, Revision 1
ESD-TR-85-149

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
The MITRE Corporation

6b. OFFICE SYMBOL *(If applicable)*

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS *(City, State and ZIP Code)*
Burlington Road
Bedford, MA 01730

7b. ADDRESS *(City, State and ZIP Code)*

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
Deputy for Acquisition (over)

8b. OFFICE SYMBOL *(If applicable)*
ALSE

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
F19628-84-C-0001

8c. ADDRESS *(City, State and ZIP Code)*
Electronic Systems Division, AFSC
Hanscom AFB, MA 01731-5000

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
64740F	522B	2522	

11. TITLE *(Include Security Classification)*
AISIM OVERVIEW

12. PERSONAL AUTHOR(S)
Schultz, Herman P.

13a. TYPE OF REPORT
Final

13b. TIME COVERED
FROM ______ TO ______

14. DATE OF REPORT *(Yr., Mo., Day)*
1985 March

15. PAGE COUNT
22

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Disrete-Event Simulation
Simulation Languages
Performance Modeling
Simulation Modeling

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

A new general purpose simulation modeling tool, Automated Interactive Simulation Modeling System (AISIM), has been developed by the Air Force System Command's Electronic Systems Division and funded by Air Force Program Element 64740F, Computer Resource Management Technology. AISIM is designed to provide quick answers to system performance questions during the conceptual phase of system acquisitions. It is currently operational on IBM and VAX computers. AISIM has undergone many tests and is now in the initial operating capability phase. Results indicate potential for wide application to the study of communications nets, computer system architectures, and data processing functions. This document introduces the reader to AISIM: its model building capabilities and its functional interfaces. Additional Keywords: interactive analysis; computer architecture; flow charting; C3I.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☒ DTIC USERS ☐

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL
Diana F. Arimento

22b. TELEPHONE NUMBER *(Include Area Code)*
(617)271-7454

22c. OFFICE SYMBOL
Mail Stop D230

DD FORM 1473, 83 APR — EDITION OF 1 JAN 73 IS OBSOLETE.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

8a. Logistics and Technical Operations.

Acknowledgments

This report was prepared by The MITRE Corporation. The work was sponsored by the Computer Engineering Applications Division (ALSE), Deputy for Acquisition Logistics and Technical Operations of the Electronic Systems Division (ESD) of the United States Air Force, Hanscom AFB, MA 01731. Funding for this report was provided by the Air Force Computer Resource Management Technology Program, Program Element 64740F (ESD/MITRE Project 5220) and ESD Computer Systems Engineering and Application project (ESD/MITRE Project 5720).

Program Element 64740F is the Air Force engineering development program to develop and employ technology, tools, and techniques essential to managing the growth in Air Force systems that use computer resources. The program's goals are to: provide for the transition of computer system developments in laboratories, industry, and academia to Air Force systems; develop and apply software acquisition management techniques to reduce life cycle costs; provide improved software design tools; address the various problems associated with computer security; develop advanced software engineering tools, techniques, and systems; support implementation of high order languages such as Ada; address human engineering for computer systems; and develop and apply computer simulation techniques for the acquisition process.

The Computer Systems Engineering and Applications Project (5720) is the ESD-initiated effort to improve the acquisition of mission critical computer resource software. The goals of the project are to: provide guidance, tools, systems, and techniques to Program Offices (including the transitioning of products from Program Element 64740F); interact with Air Force and DOD organizations that establish policies, regulations, and standards for software acquisitions; and direct associated technology efforts.

Contents

Figures

Section One

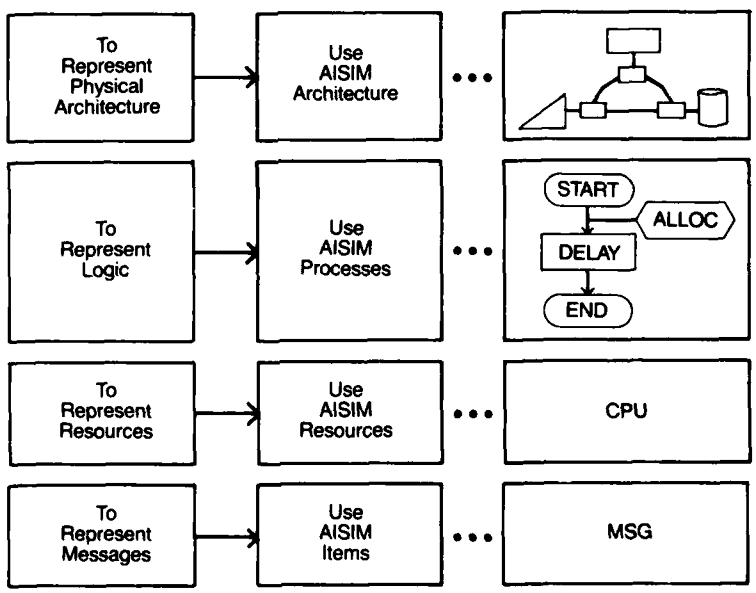
Introduction

Automated Interactive Simulation Modeling System (AISIM) is an interactive, graphics-oriented modeling system capable of simulating a variety of problems ranging from user-machine interfaces to distributed computer systems. It uses interactive graphics to specify a system's architecture and to describe the functional processes that operate within that architecture. AISIM produces plots of performance statistics for messages, system resources, and system functions. Reports include queuing (time in queue, queue length), resource usage, and message transit time statistics. Flowcharts of a model's logical processes are automatically produced providing an excellent source of model documentation. AISIM runs under the IBM MVS/TSO and the VAX VMS operating systems.

AISIM is designed to be particularly useful during the conceptual design phase of Command, Control, Communications, and Intelligence (C^3I) system acquisitions. It is useful for the design and analysis of computer-based systems including local area networks, data processing systems, and distributed systems. It can be used to perform sizing and timing studies in the development of software and hardware designs, system architectures, and protocols. AISIM supports the preparation of specifications by quickly providing answers on the technical feasibility and impact of performance requirements. It can also be used to validate the performance of proposed designs. While a system is being built or after it becomes operational, AISIM can be used to model proposed changes and enhancements in order to determine cost-effective solutions.

AISIM is an easy-to-learn tool that can be used directly by an analyst, thus eliminating the time-consuming and error-prone interaction between analyst and simulation programmer. AISIM's high-level representative language facilitates the preparation of a model. AISIM modeling constructs correspond to those of the system being modeled, resulting in a mapping relationship between AISIM entities and those of the modeled system (figure 1).

Figure 1
Mapping of System into AISIM



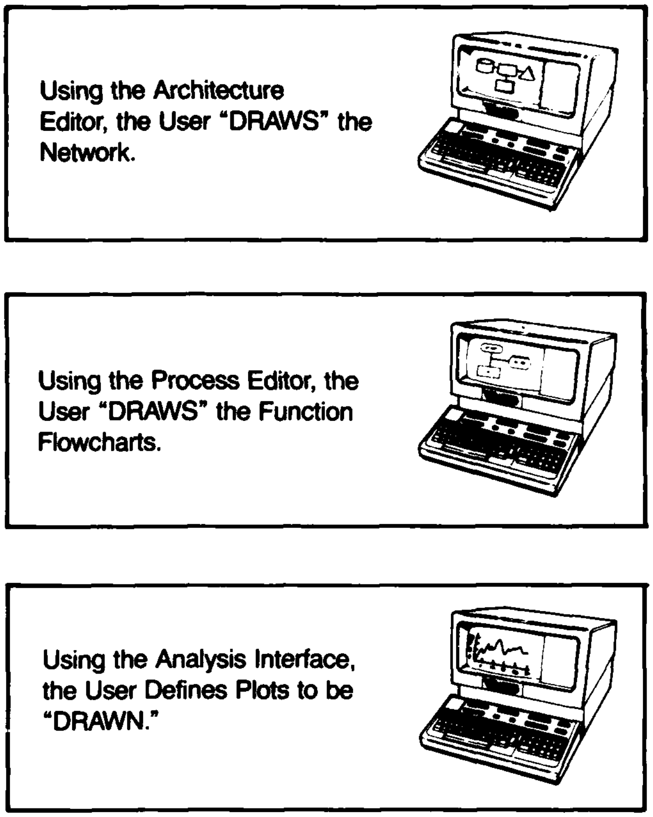


Figure 2
Interactive Graphics

A network's physical architecture is represented by using AISIM's architecture design feature. The physical resources of the modeled system are depicted graphically on the terminal screen. The logical processes of a system are represented by AISIM entities called processes. Processes contain elements called primitives that correspond to steps in the modeled system's logic. The hardware resources of the modeled system are represented one-for-one by AISIM entities called resources. AISIM also provides items that represent transient data elements, such as messages, that can flow through a modeled network. This correspondence between system components and AISIM entities simplifies the model builder's task.

AISIM's interactive graphics assists the analyst in several modeling areas. Of the three examples shown in figure 2, the first illustrates AISIM's capabilities that enable the user to "draw" the modeled architecture on the screen. The second shows model processes represented in the form of a flowchart called a process diagram. As the user builds a process, this flowchart is drawn automatically. The third example illustrates AISIM's plot output. AISIM provides statistical plots for performance attributes of system resources and message transactions. Performance measures such as throughput, response time, resource usage, and queuing may be viewed as plots at the completion of a simulation.

These are only a few of the capabilities made possible by AISIM's five functional interfaces. The next section describes each of these interfaces, and the final section gives examples to illustrate AISIM's model building and execution features.

Section Two

AISIM Interfaces

AISIM consists of five functional interfaces as shown in figure 3: Design, Analysis, Replot, Hardcopy, and Library. The Design Interface (DI) provides the user with the commands necessary to design and construct a model. The model is actually run or executed using the Analysis Interface (AI), which provides the commands necessary to control the simulation and view simulation results. The Replot Interface (RI) allows the user to view plots from different runs and to plot them on the same graph. The two remaining interfaces, the Library Interface (LI) and the Hardcopy Interface (HI), provide tools for model documentation and for saving models to be used in subsequent studies.

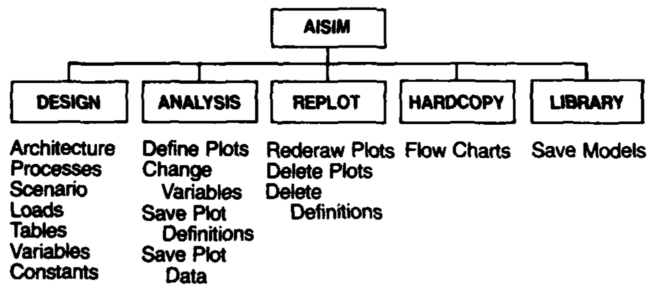


Figure 3
AISIM Interfaces

Design Interface

In creating a model, the DI is used to describe the modeled system's functions and architecture through a set of constructs called entities (table 1). The DI is used to select and enter the required entities into a model. It contains commands for creating and modifying model elements including the architecture, the processes that describe the model logic, and the external system loads. Other model entities include model resources not defined in the architecture, and entities such as tables, constants, and variables.

Most entities are created using the edit command. This command causes a form to be displayed for the entity being defined. When a form is completed and entered, the defined entity becomes part of the model. Forms help the modeler enter data correctly and ensure that all necessary data is included. Forms also facilitate quick model building and reduce the probability of errors being introduced into the model. Creation of the model architecture and model processes are complex functions that make use of graphics as well as forms. These and the means used to define the load on the modeled system are discussed below.

Architecture Editor (AE) The AE is used to create the system architecture in graphic form (figure 4). It displays the layout and interconnections of the physical elements of the modeled system. Arbitrarily chosen symbols represent physical system elements. The lines or links connecting them represent communications paths.

ACTION	Used to tabulate statistics on the time a resource was used for a particular function.
CONSTANT	A model parameter whose value cannot be changed during the simulation.
ITEM	Used to represent transient data elements, such as messages.
LEGAL PATH TABLE	Created from the modeled architecture; defines routes through the system.
PRIMITIVE	Represents a logical step of a process. A sequence of primitives compose a process.
PROCESS	A description (using primitives) of the logic of a function being modeled.
LOAD	Used to schedule processes at a repetitive rate to generate transactions (messages) for the system. Load triggering is specified in the scenario entity.
SCENARIO	Specifies the length of time being simulated and schedules loads/processes.
RESOURCE	Used to represent the actual resources (CPU, disk, channel, terminal, etc.) of the system being modeled.
VARIABLE	A model parameter whose value can be changed both before and during the simulation.

Table 1
AISIM Entities

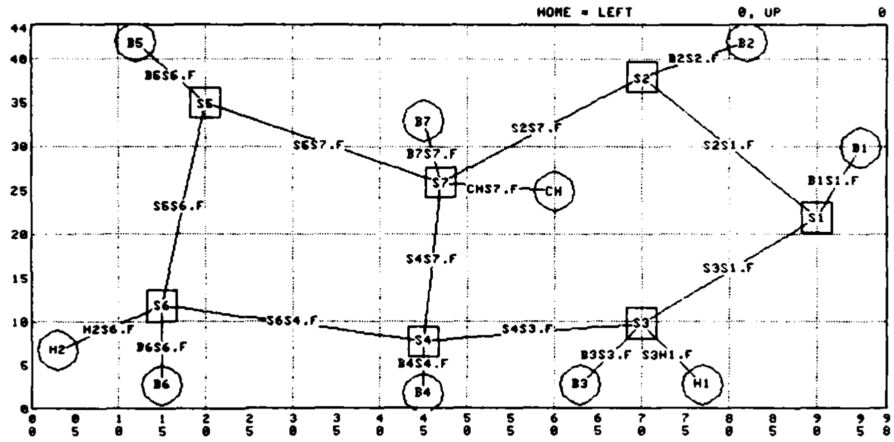


Figure 4
Architecture Example

ARCHITECTURE LEGAL PATH DEFINITION

FROM DEVICE	TO DEVICE	NEXT DEVICE	VIA LINK
B1	B1	S1	B1S1.A
B1	B2	S1	B1S1.A
B1	B3	S1	B1S1.A
B1	B4	S1	B1S1.A
B1	B5	S1	B1S1.A
B1	B6	S1	B1S1.A

Figure 5
Legal Path Table

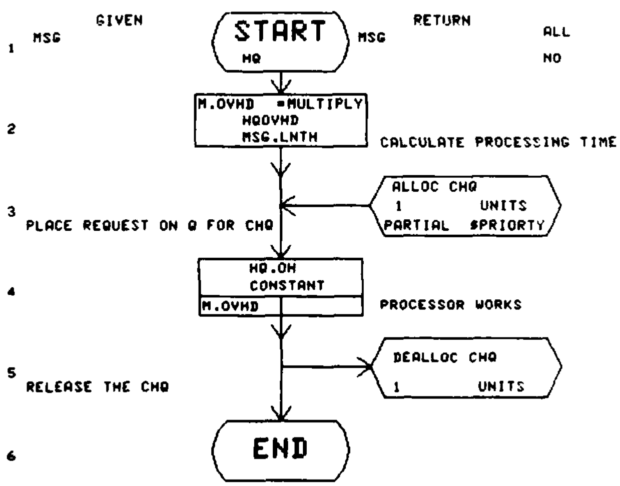


Figure 6
Process Flow Diagram

For each node and link, AISIM creates a resource entity to which attributes may be assigned. The AE can also create a Legal Path Table (LPT), sometimes referred to as a routing table, from the architecture diagram. Figure 5 shows part of the LPT for the architecture of figure 4. The LPT contains the paths or routes to be used between each pair of nodes. An entry in the table consists of four elements: the FROM node, the TO node, the NEXT node on the path, and the next LINK on the path. The LPT is used by model processes to route messages through the modeled system.

Process Editor (PE) The functions of a modeled system are described in AISIM by processes. The PE creates and modifies processes and displays the elements of a process as a flow diagram (figure 6). Each functional element is called a primitive (table 2). Easy to learn and use, the primitives are the only "language" the user must learn in order to build models in AISIM.

Scenario and Load Entities The scenario and load entities are used to define the model environment in the form of information and data passed to it. The scenario (figure 7) allows the analyst to specify the number of time units per simulation period and the number of periods in the simulation. Breaking the simulation run into periods provides the analyst with the flexibility to examine intermediate results. The scenario is also used to specify one or more loads, and the time and priority to trigger each.

The load (figure 8) permits processes to be scheduled cyclically or repetitively. It is used to schedule processes that create items (messages) which represent external loads or inputs to the system. The load identifies the rate at which processes will be scheduled and items generated. Each time a process is scheduled a new instance (copy) of the process is created and executed. Many instances of the same process, therefore, may be executing simultaneously, though perhaps at different stages of their logic. The same process may also execute at different nodes in the modeled system. Thus, several instances of a particular process may exist at every node in a modeled network simultaneously. But it is only necessary to define the process once.

After a system model is defined by specifying its processes, scenario, loads, resources, legal path table, constants, variables, and so on, the model is ready to be exercised using the AI.

Analysis Interface

Using the AI, the analyst typically defines plots and system variables, runs the simulation, and views results in the form of printed statistics or plots (figure 9). Several runs can be made quickly to determine how the model responds to various loading conditions and/or capacity parameters. Plots for a particular resource or message type are defined by choosing from a table entitled "Attributes" and from a subsequently displayed table entitled "Statistics." An example of a resulting plot is also shown in the figure. At the completion of a run, plots and statistical reports may be viewed on the screen and/or printed. The AI also enables the analyst to save plots for subsequent analysis and use.

Replot Interface

The RI allows the user to retrieve and display plots that were saved in the AI. With the RI, plots from several different runs can be selected and viewed individually or overlaid on the same graph for comparison purposes. The RI also provides a housekeeping function for plots and plot definitions by providing a facility for deleting them when they are no longer needed.

Library Interface

The LI enables the user to save models or selected model processes on either a user or system library from which they may be retrieved and used to develop new models. The user's library is controlled by the user, who may add or delete entries as desired. The system library is accessible by the user, but only the system librarian is able to modify its contents.

A set of AISIM processes called the Message Routing Submodel (MRS) resides on the system library. These processes may be inserted into a model to perform message routing functions. The MRS uses

CALL **SEND** **SUSPEND** **RESUME** **WAIT**	Control execution and suspension of processes.
CREATE **DESTROY**	Generate and eliminate messages.
ACTION	Consumes time.
EVAL **ASSIGN**	Assign values to variables and perform mathematical and statistical functions.
ALLOC **DEALLOC** **RESET** **LOCK** **UNLOCK**	Changes status of resources.
FILE **FIND** **REMOVE**	Manipulate special user-defined queues.
BRANCH **COMPARE** **ENTRY** **LOOP** **PROB** **TEST**	Control flow of process logic.
TRACE	Debugging.

Table 2
AISIM Primitives

node and link resources while routing messages through a network. It can also schedule a user-defined function at a message destination that could, for instance, generate and route an acknowledgment or response to the originator. As with any library entry, the MRS may be used as is, or it may be modified by the user.

Hardcopy Interface

The HI provides the analyst with the capability of producing hardcopies of the process flow diagrams. This interface provides the commands necessary to print the desired output on a graphics printer.

The HI is an invaluable tool that aids in the documentation of the completed model. In addition, it facilitates the maintenance of model documentation. As processes are modified the process diagrams are automatically updated, and revised hardcopies may be produced using the HI.

Form Input

A special feature of AISIM is its use of forms for data entry. Each form uses labels to define its fields. The fields are highlighted with the "inverse-video" feature of the terminal. As an additional

```
SCENARIO DEFINITION

SCENARIO
MNEMONIC          DESCRIPTION
======== ======== ==================================================
RUN

PERIOD
LENGTH
========
3600

         PERIOD   PERIOD   PERIOD   PERIOD   PERIOD   PERIOD   PERIOD
         MNEMONIC MNEMONIC MNEMONIC MNEMONIC MNEMONIC MNEMONIC MNEMONIC
======== ======== ======== ======== ======== ======== ======== ========
         ONE

TRIGGER  TIME  TO SCHEDULE TRIGGER  TIME  TO SCHEDULE
MNEMONIC SCHEDULE PRIORITY MNEMONIC SCHEDULE PRIORITY
======== ======== ======== ======== ======== ========
HOUR     0        1
```

Figure 7
Scenario Definition Form

```
LOAD DEFINITION

LOAD
MNEMONIC          DESCRIPTION
======== ======== ==================================================
HOUR              AN HOUR OF MESSAGES

         LOAD     NODES
         ======== ======== ======== ========
         NODE1


         PROCESS           SCHEDULE
         MNEMONIC MAX #    METHOD   MEAN     DELTA    PRIORITY
         ======== ======== ======== ======== ======== ========
         SEND-MSG 180      EXPONENT 18.7              1
         DATABCHQ 125      EXPONENT 25                0
```

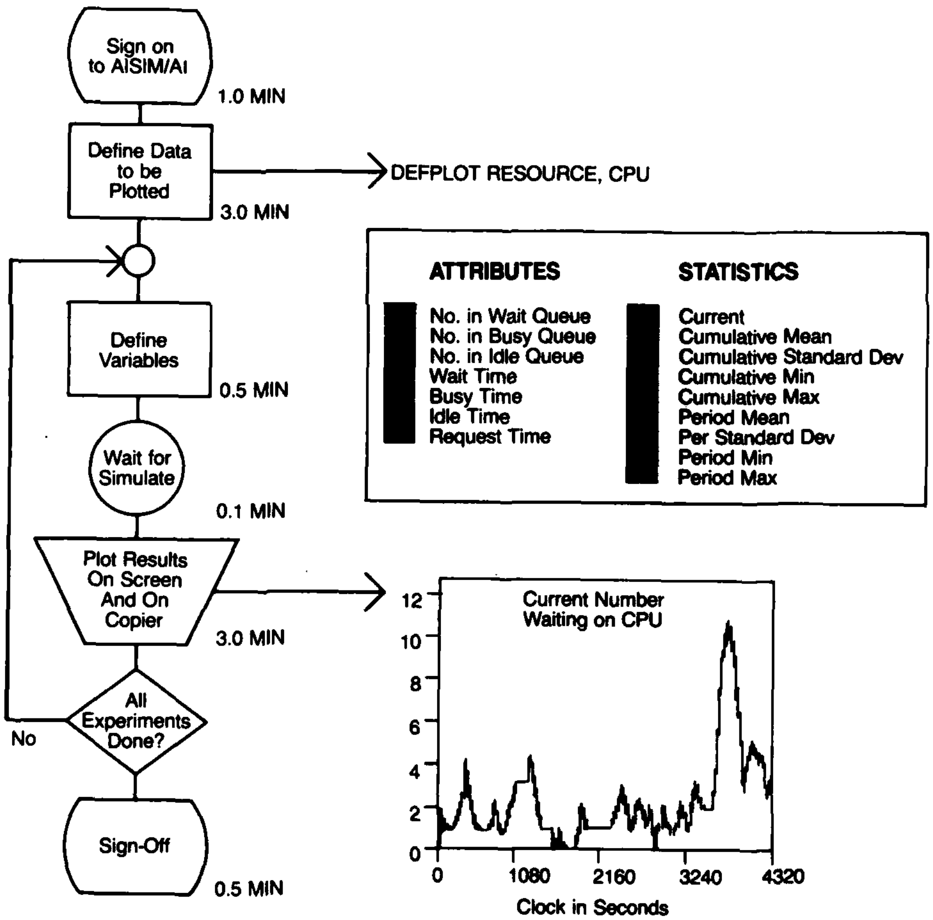
Figure 8
Load Definition Form

safeguard, data can be entered only into the highlighted fields. Forms help the modeler enter data correctly and ensure that all necessary data is included. They facilitate quick model building and reduce model debugging time.

Summary

The simple, powerful functional interfaces of AISIM enable an analyst, with essentially no knowledge of programming, to quickly and efficiently develop system performance models. The AISIM commands are easy to learn and permit rapid model building and analysis. Experience indicates that a typical user can learn AISIM within one to two weeks, then start to build, execute, and analyze models.

Figure 9
Model Analysis Cycle



Section Three

AISIM Use

This section gives representative examples of how AISIM is used to model command, control, and communications system functions. The first part discusses modeling examples using the DI, and the second part illustrates how the AI is used for execution and analysis of models, including examples of plots and statistical summaries.

Design Interface

The DI is entered when the user wishes to create or modify a model. Examples of the three major aspects of AISIM model design (architecture definition, function specification, and system load definition) follow:

Architecture Definition The Architecture Editor (AE) of the DI is used to define a system's physical architecture, that is, the layout and interconnections of the system's physical resources: CPU, disk drive, tape drive, channel, and so forth. The user, employing the terminal's graphic features, creates a graphic picture of the system architecture by positioning symbols and connections between them. The 14 symbols provided by AISIM, together with their mnemonic labels, are shown in figure 10. The analyst also may assign attributes and attribute values to specific symbols to characterize physical devices. At the end of an AE session, a Legal Path Table can be generated either automatically by AISIM, or manually by the user. This table defines paths between pairs of nodes in the architecture, and is referenced by model processes in routing messages through the modeled system.

Figure 11 is an example of a loop architecture created using the AE. In this network, messages are generated at hosts (H1, H2, . . . H6) and sent to Loop Interface Units (LIUs) represented by B1, . . . B6. The messages are then transmitted via the loop and routed to the next LIU. After a processing delay each message is routed either to the attached host or the next LIU, depending on its destination.

In this example, AISIM automatically generates resources for the six hosts, the six LIUs, and the 12 channels defined by the user in the architecture diagram. Full duplex channels are specified by appending ".F" to the channel name. Thus, the channels connecting the hosts and the LIU's are full duplex channels. The user may define and assign a set of attributes to a resource. For instance, if the delay of messages at the LIUs is of interest, one of the attributes associated with the LIU resource could indicate the processing delay per message character. If this value is defined as a global constant or variable it can easily be changed at the beginning of a simulation run. This delay value could then be used by a process to calculate LIU processing delays during the

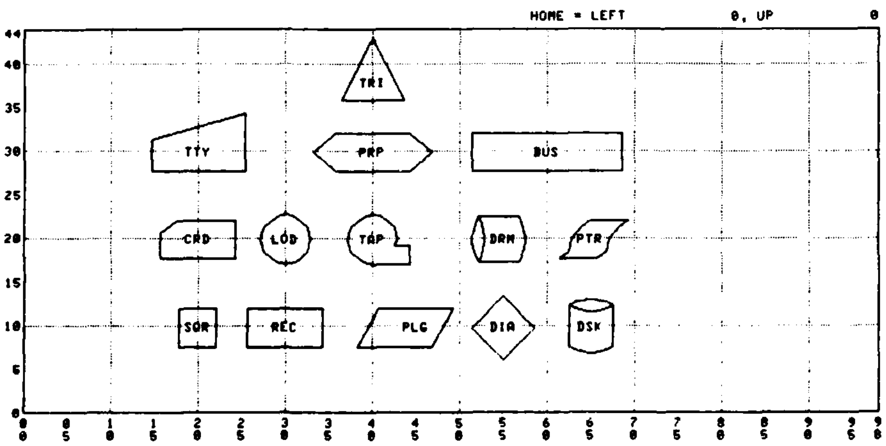
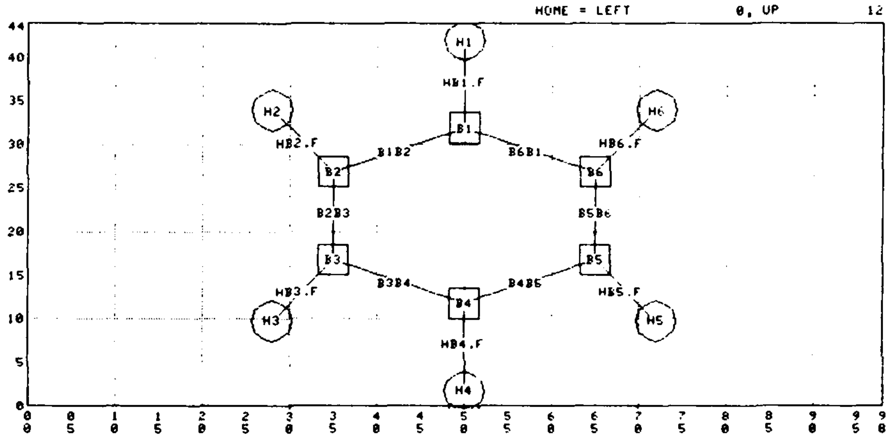


Figure 10
Architecture Design Symbols

Figure 11
Loop Architecture
Example



course of the simulation. Subsequently, statistical information on messages could be compiled automatically during the run and viewed at a later time.

If the model requires an LPT, the user may choose to have AISIM generate the LPT at the end of the AE session, or may define communication paths using AE commands. If AISIM generates the LPT and ambiguities are encountered, AISIM will query the user to determine the desired path.

Each entry in the LPT consists of a FROM node, a TO node, a NEXT node, and a CHANNEL. The LPT is referenced in the model processes with system keywords. Given the current node and the destination node of a message, for example, the LPT is then referenced with the appropriate keyword to obtain either the next node along the message path ($NXTNODE) or the channel needed to reach that node ($CHANNEL).

Function Specification The Process Editor (PE) is entered from the DI level when the analyst wishes to create or modify a process. A process defines a function or set of functions of the modeled system. Each process is composed of primitives that define the process functions. It is at this level that resources are allocated and deallocated, time is consumed, decisions take place, and so on. Since resources are limited and shared, AISIM automatically maintains statistics on timing and resource contention factors during model execution. Processes are initiated by scenarios, loads, and by other processes. Once initiated, execution of a process may depend on the availability of resources. Processes can represent communications functions as the following examples illustrate.

Example I Messages are generated at several stations and transmitted to a master station. The purpose of the model is to obtain queuing statistics for the transmission lines and the receiving processor, and to obtain statistics on message transit times.

One process required to model this system defines the message reception and processing functions at the master station. To create this process, the user enters the EDIT PROCESS command specifying the process' name. The process' parameters are then entered in the process form displayed by AISIM (figure 12). The process name in this example is REC-MSG and the master station node is

```
START

PROCESS NAME  REC-MSG   NODE  NODE2

ATTRIBUTES ATTACHED (YES OR NO)  NO

PROCESS DESCRIPTION

RECEPTION AND DESTRUCTION OF MESSAGE

START BLOCK TYPE  PARM

ENTER "PARM" FOR PARAMETER PASSING
ENTER "ITEM" FOR ITEM PASSING
ENTER "STD" FOR STANDARD PROCESS
```

Figure 12
Process Form

called NODE2. Since this process must receive message items before operations can begin, it is designated as a parameter passing process by entering "PARM" in the "START BLOCK TYPE" field. Figure 13 shows the additional form displayed by AISIM for a parameter passing process. In this case, the user enters the name "MSG", a pointer to the message being passed to this process. AISIM then draws the basic structure of the process (figure 14).

As the user develops the process by adding primitives, AISIM redraws the process flowchart. The appropriate primitives are placed in the flowchart using the PE commands. For example, to generate the first operation after the START, the command PLACE ASSIGN is entered. AISIM displays a form for the ASSIGN primitive (figure 15) in which the user enters parameters. This example places the LENGTH attribute of MSG into a local variable called MSG.LENG. The fully developed process (figure 16) receives a message, computes processing time based on the message length, allocates the resource required, uses the resource for the time required to process the message, destroys the message, and deallocates the resource.

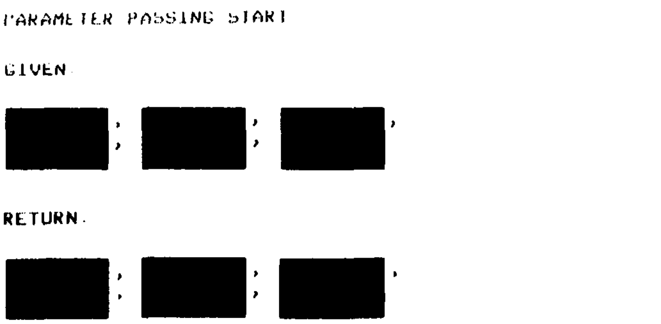


Figure 13
Parameter Form

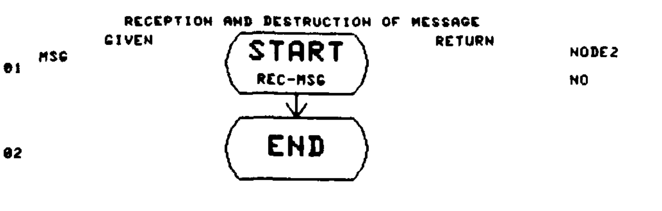


Figure 14
Initial Process Flow Diagram

The entities addressed or manipulated by the primitives of a process (resources, constants, variables, items, etc.) are also defined by the user. For instance, in the process just defined, an instance of an item called MSG must have been created and passed to this process. To define the format or template for an item the EDIT ITEM command is used. The item form (figure 17) shows the information defining MSG entered in the appropriate fields. To create an instance of MSG, the CREATE primitive is used in a model process. Figure 18 shows the form used for defining a resource. For resource NODE2, one unit of the resource is available at the start of the simulation. The number of units available during the simulation may be changed by the process logic. This feature may be used to simulate a resource failure by making the number of units available equal to zero.

Example 2 This problem, which requires modeling of a bus communications network (figure 19), is typical of a class of problems that deal with multiple hosts communicating over a local network. In this example H1, . . . H4 are connected to the bus through Bus Interface Units (BIUs) represented by B1, . . . B4. Messages generated at each host are routed to other hosts via the bus. Queuing and usage statistics are to be generated for the bus, channel, and processor resources. Message input traffic consists of data requests generated randomly by the hosts.

The process shown in figure 20 models the BIU functions. The BIU receives data request messages from the databus, generates acknowledgments for each message, and transfers these messages to the attached host for processing. The process is designated a parameter passing process because it is to be called from another process and passed a pointer to a message (MSG). The GENACK and SENDACK processes are then called to generate and send the message acknowledgment. Primitives 4, 5, and 6 in the process obtain the name of the BIU-to-host channel using the Legal Path Table (LPT). Primitive 6 uses a system keyword ($CHANNEL) that references the LPT to determine the next channel over which the message is to be transferred given the current node ($CNODE) and the destination node (TO.NODE). This channel is then allocated and primitive 8, an ACTION primitive, specifies the amount of time the channel is used — in this case 3 msec. The channel is then de-allocated and process HOST is called to perform message processing.

System Loads The external inputs to a system are modeled using the scenario and load entities. The scenario entity specifies the length and number of periods of the simulation. Periods allow the user to stop the simulation to alter parameters or inspect results. The scenario triggers a number of loads and/or processes, specifying the scheduling time and priority for each. Load entities specified in a scenario trigger processes according to the scheduling method specified. A load is normally used to schedule processes that create items (messages). The scheduling method may be any one of

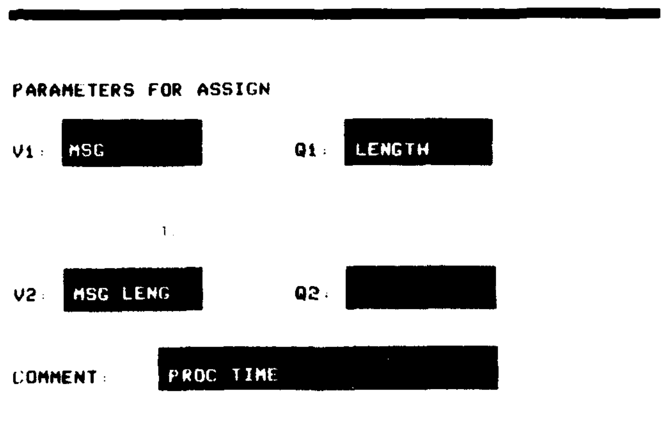


Figure 15
Form for ASSIGN Primitive

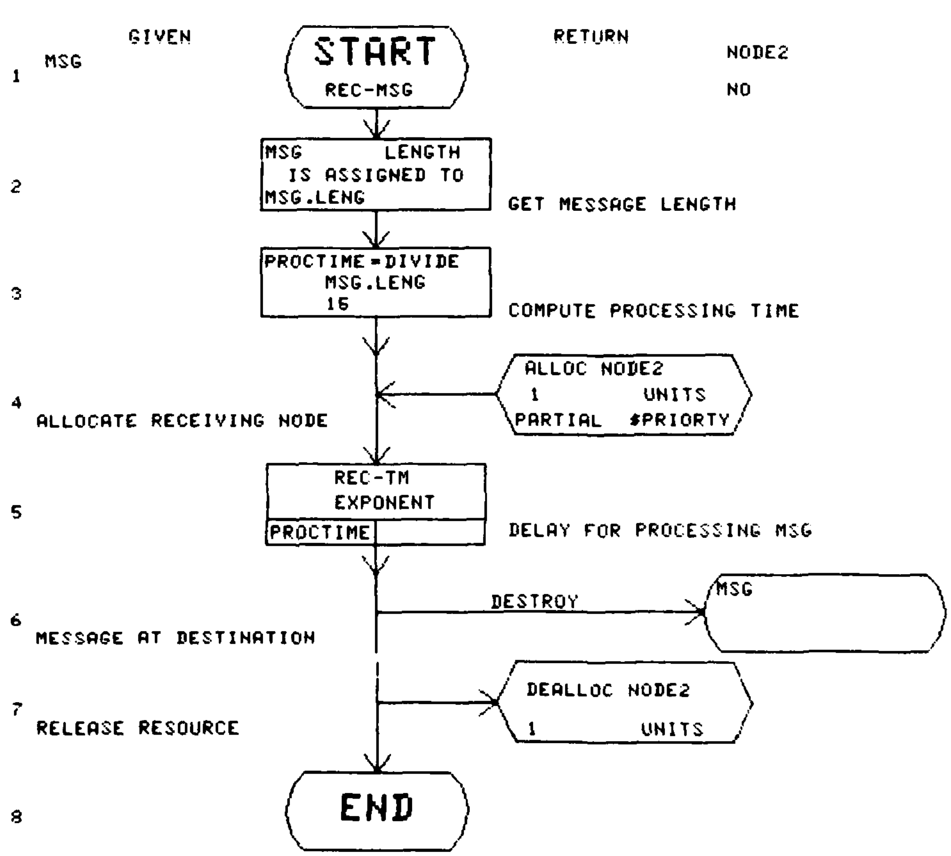


Figure 16
Complete Process Flow Diagram

12 statistical distributions, including Poisson and exponential. The load also specifies the node or nodes in which the scheduled process is to operate.

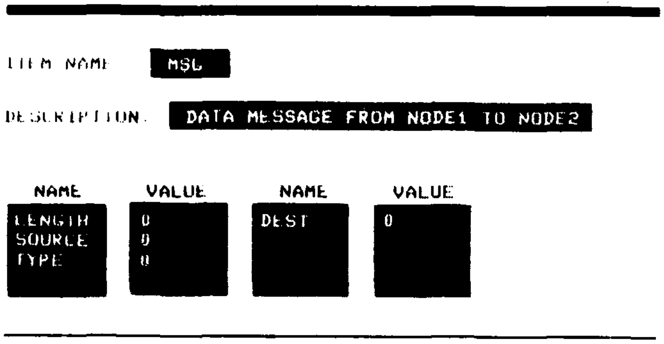


Figure 17
Item Definition Form

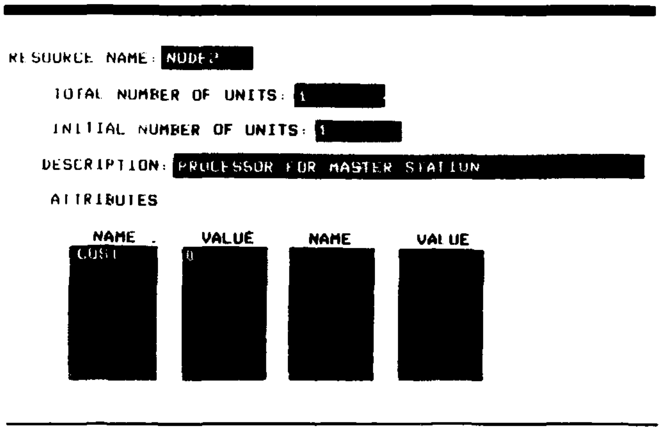


Figure 18
Resource Definition Form

For the bus communications network (example 2), message input traffic for the modeled network consists of data requests generated by the hosts. Data requests are generated randomly using an exponential distribution (Poisson arrival pattern).

The scenario for this example (BP2SCEN) is shown in figure 21. One period is specified with a length of 1,000,000 msec. The unit of time for this model was chosen by the modeler to be in milliseconds. The scenario also specifies that loads LOADH1, . . . LOADH4 are to be scheduled at time zero with a priority of zero. The load LOADH1 is shown in figure 22. The three processes specified in the load create messages for specific destinations. For example, TOHOST2 creates a message with destination H2. Thus, the load specifies that a maximum of 200 messages destined for H2 is to be generated with the time between messages exponentially distributed around a mean of 101,010 msec.

Analysis Interface

AISIM models are executed using the AI. If more than one scenario has been defined, the AI asks the user for the scenario to be used. Once the scenario is selected, AISIM translates the model definition file into a serial input file and error-checks this file. If errors are encountered a message is printed. AISIM identifies errors such as inconsistency in data types, unresolved addresses, and undefined terms. A file is listed containing the flagged errors.

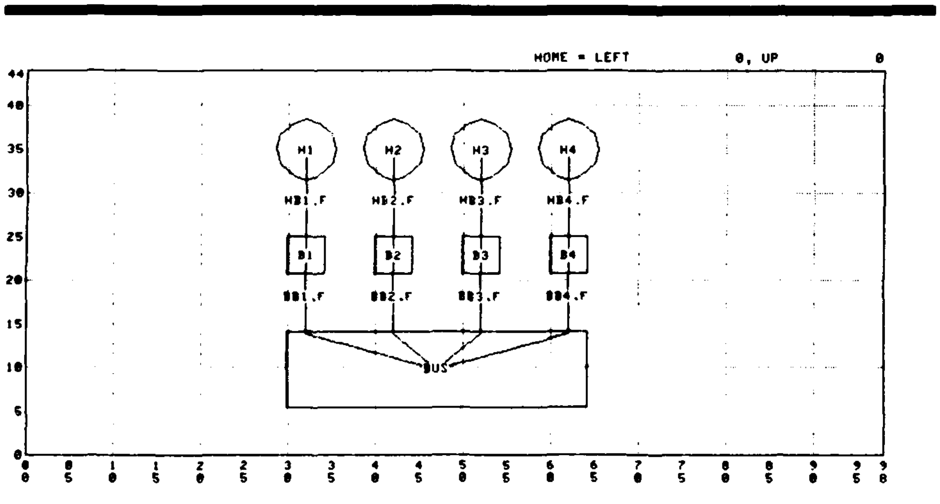


Figure 19
Bus Architecture

Commands that control the simulation may be used both before and during model execution. Prior to execution the user may specify simulation data to be plotted, modify variables of the model, and define breakpoints in the execution. During execution the user interfaces with the model at breakpoints or ends of periods that are defined before execution. At these times data may be plotted, parameters or, breakpoints changed, and so forth. At the end of execution data may be plotted and a listing obtained of all statistics produced by the model.

Statistical Summaries Standard statistical summaries are produced automatically by AISIM for each simulation. During model execution AISIM automatically collects statistics that include resource usage (time and percent), queue times and lengths, transmission times, and process timing. Among the statistical reports produced are a resource report and an item report.

AISIM produces a resource report for each resource used in the simulation. During model execution AISIM maintains four queues: idle, busy, inactive, and wait. The first three queues are really "states" of the resource; the fourth is the actual queue for the resource. During the simulation the idle queue contains the number of resource units that are available but unallocated, the busy queue contains resources that are allocated, the inactive queue contains resources that are unavailable, and the wait queue contains processes waiting for the resource.

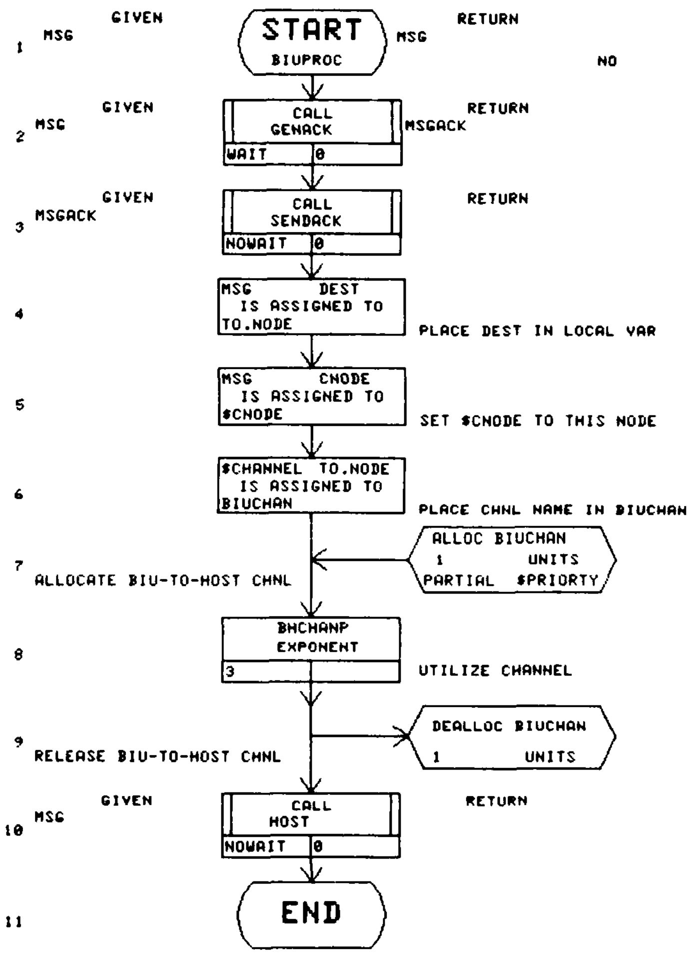
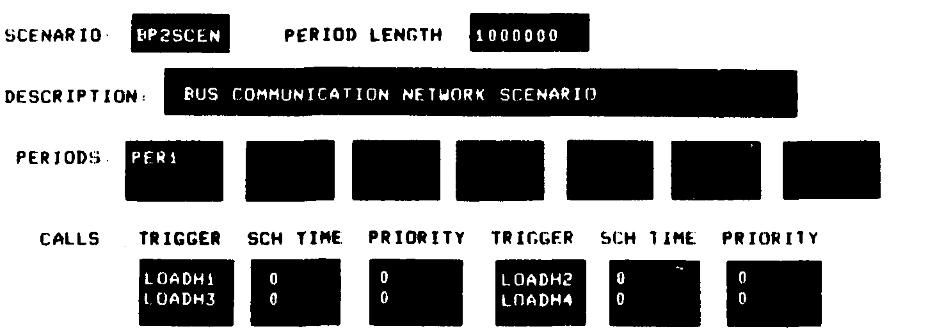


Figure 20
BIU Data Request Process

Figure 21
Scenario Definition Form



LOAD: LOADH1

NODE 1	NODE 2	NODE 3	NODE 4

NODE 5	NODE 6	NODE 7	NODE 8

DESC: LOA[illegible] HOST H1

PROCESS	MAX #	SCHMTD	MEAN	DELTA	PRIORITY
TOHOST2	200	EXPONENT	101010		0
TOHOST3	400	EXPONENT	101010		0
TOHOST4	400	EXPONENT	50505		0

Figure 22
Load Definition Form

RESOURCE REPORT

RESOURCE	TOTAL NUMBER	CURRENT	MEAN	STD DEV	MINIMUM	MAXIMUM
CHANNEL						
# IDLE		1 000	0 337	0 473	0	1 000
REQUEST TIME			175814	22 884	0	104 975
INTO BUSY	180					
OUT OF BUSY	180					
# BUSY		0	0 663	0 473	0	1 000
BUSY TIME			13 253	13 912	109	91 170
# INACTIVE		0	0	0	0	0
INTO WAIT	180					
OUT OF WAIT	180					
# WAITING		0	0 893	1 284	0	6 000
WAIT TIME			17 851	22 884	0	104 975

CURRENTLY ALLOCATED TO PROCESSES NONE

PROCESSES CURRENTLY WAITING NONE

Figure 23
CHANNEL Resource Report

ITEM REPORT

ITEM NAME	NUMBER CREATED	NUMBER DESTR'D	TIME IN SYSTEM MINIMUM...	MAXIMUM...	AVERAGE...	STD DEV...
MSG	180	180	.65	183.57	53.17	38.73

Figure 24
Item Report

Figure 23 shows a resource report for a resource called CHANNEL. The resource CHANNEL may be a link over which messages created at a transmitting node are sent to a receiving node. The report identifies the resource attributes in the row headings and the individual statistics in the column headings. The report shows that the resource was put into the busy state 180 times, which corresponds to the number of messages created. Since there is only one unit of the resource CHANNEL, the mean # busy indicates that the channel was used 66.3 percent of the time for message transmission. The maximum # waiting indicates that the maximum queue length for this resource was 6 and the maximum wait time shows that the worst case wait time was 104.975 msec.

The item report provides statistics on each item (message type) used in the model. Figure 24 is an example of an item report for a model that had only one item definition: MSG. The report shows the number of items that were created and destroyed, as well as the minimum, maximum, and average time an item was in the system. In this example the number of items called MSG that were created and destroyed was 180. The statistics on a message's time in the system are a function of the processing times for the message and the wait times for the resources needed. For this model the average time in the system for items called MSG was 53.17 msec; the maximum was 183.57 msec.

Plot Output Before execution the user may request plots of various statistical data. These plots can be displayed and printed at the end of the simulation or at the end of a simulation period.

Figure 25 shows the forms used to request a plot for resource statistics. The "Attributes" form indicates the data to be addressed and the "Statistics" form indicates the value desired.

Three examples indicate the kinds of data that can be plotted. The current number of processes waiting for the resource CHANNEL is plotted in figure 26. This plot gives the user statistics on the dynamic activity of the wait queue for this resource. Figure 27 shows plots of both the current time and cumulative mean time in the system for items called MSG. Figure 28 shows the number of MSG items in the system at any given time.

System Modeling Examples

"AISIM Evaluation — Preliminary Report"[1] contains five examples of AISIM's use for communications system modeling. The problems modeled were based on ESD system acquisitions and were used in the evaluation and testing of AISIM.

Summary

AISIM is a highly interactive discrete-event simulation modeling system. It provides a graphics capability for describing a modeled system's architecture and functions, a simulation capability for analyzing the system, a reporting system for obtaining performance measures, and a database for storing model designs and definitions. AISIM'S

ATTRIBUTES	STATISTICS
No. in Wait Queue	Current
No. in Busy Queue	Cumulative Mean
No. in Idle Queue	Cumulative Standard Dev
Wait Time	Cumulative Min
Busy Time	Cumulative Max
Idle Time	Period Mean
Request Time	Per Standard Dev
	Period Min
	Period Max

Figure 25
Plot Definition Forms for Resources

Figure 26
Wait Queue for Resource CHANNEL

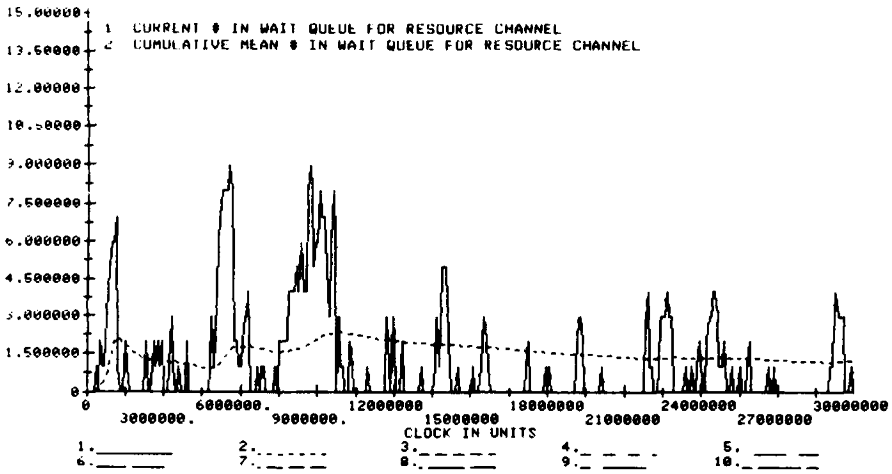
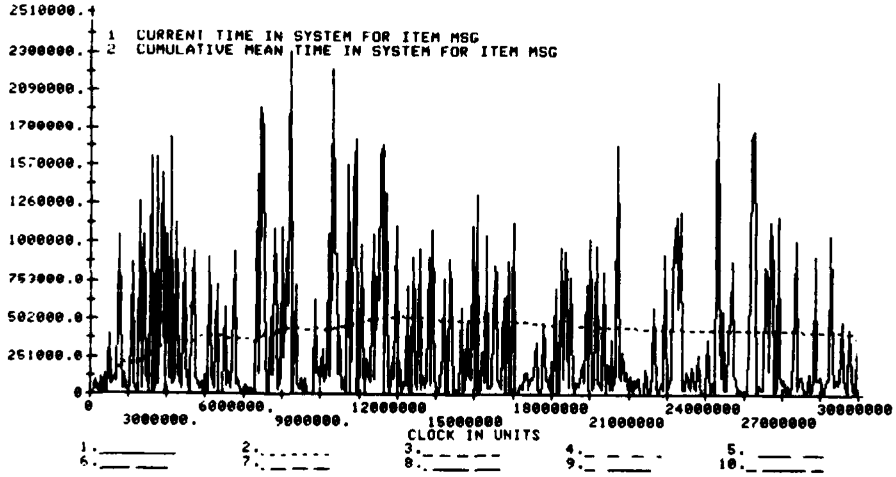


Figure 27
Time in System for Item MESSAGE



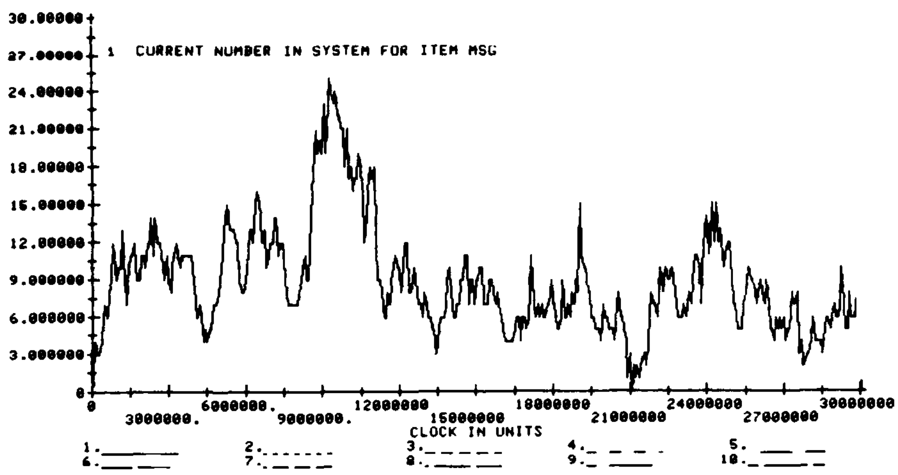


Figure 28
Number in System for Item MESSAGE

interactive organization assists the user in several modeling areas. It uses interactive graphics to enable the analyst to "draw" the modeled architecture on the screen and to describe the functional processes that operate within that architecture. Each modeled process is represented as a flowchart that is updated and drawn automatically. The architecture diagrams and process flowcharts provide a ready source of model documentation. AISIM automatically produces standard statistical summaries and allows the user to request plots of performance statistics. The library feature allows the user to save model designs and definitions in a library, and retrieve them or portions of them for developing new models.

AISIM is an easy-to-learn tool designed for direct use by an analyst as a workbench for investigating performance impacts of design alternatives in a short period of time. It is particularly useful during the conceptual phase of command, control, communications, and intelligence system acquisitions. Experience to date indicates that AISIM has vast potential for application to a wide range of modeling problems.

[1]H. P. Schultz, S. Natarajan, and J. K. Fryer, "AISIM Evaluation — Preliminary Report," ESD-TR-82-119, Electronic Systems Divison, AFSC, Hanscom AFB, MA, Sept. 1981, AD A 113108.

END

FILMED

9-85

DTIC